スタートガイド

QEC-M-01 Modbus TCP 制御
デジタル I/O

86Duino Coding IDE 501

Ethernet ライブラリ & Modbus ライブラリ

（バージョン 1.0）



改訂履歴

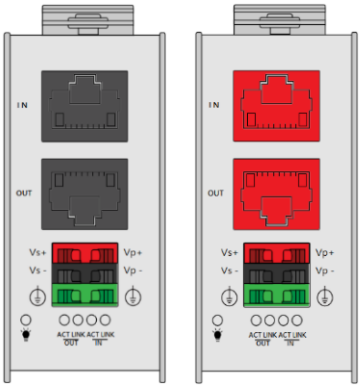
日付	バージョン	内容
2025/06/12	Version 1.0	初版リリース。

はじめに

本ガイドでは、EtherCAT M デバイス QEC-M-01 から Modbus ライブラリを使用して Modbus TCP デジタル I/O を制御する方法を説明します。

注意：QEC の PoE（Power over Ethernet）について

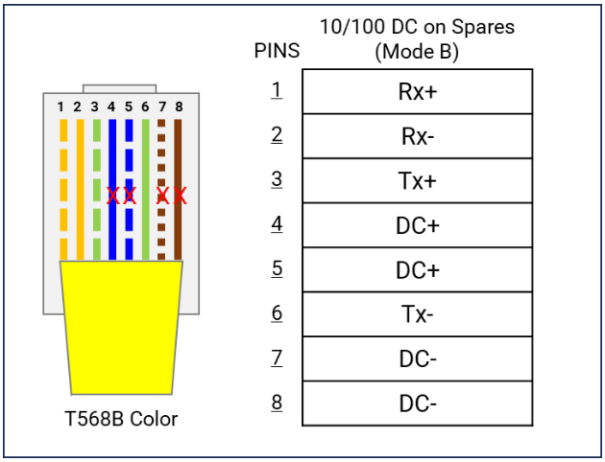
QEC 製品の設置時には、PoE タイプと非 PoE タイプを簡単に見分けることができます。RJ45 コネクターのハウジングが赤色の場合は PoE タイプ、黒色の場合は非 PoE タイプです。



非 PoE タイプ　　PoE タイプ

PoE（Power over Ethernet）は、ネットワークケーブルを介して電力を供給する機能です。QEC ではオプションで PoE 機能を搭載でき、配線を削減できます。実際の運用では、システム機器に応じて PoE を選択するため、評価・テストの際は以下の点にご注意ください。

1. QEC の PoE 機能は EtherCAT P とは異なり、互換性はありません。QEC の PoE 機能は PoE Type B に準拠しており、ピン機能は以下のとおりです。



2. PoE 対応機器と非対応機器を接続する場合は、必ずイーサネットケーブルの 4、5、7、8 番ピンを切断してください（例：PoE 対応の QEC EtherCAT M デバイスとサードパーティ製の EtherCAT Sub デバイスを接続する場合）。

3. QEC の PoE 給電能力は最大 24V/3A です。

1. ハードウェアの接続と配線

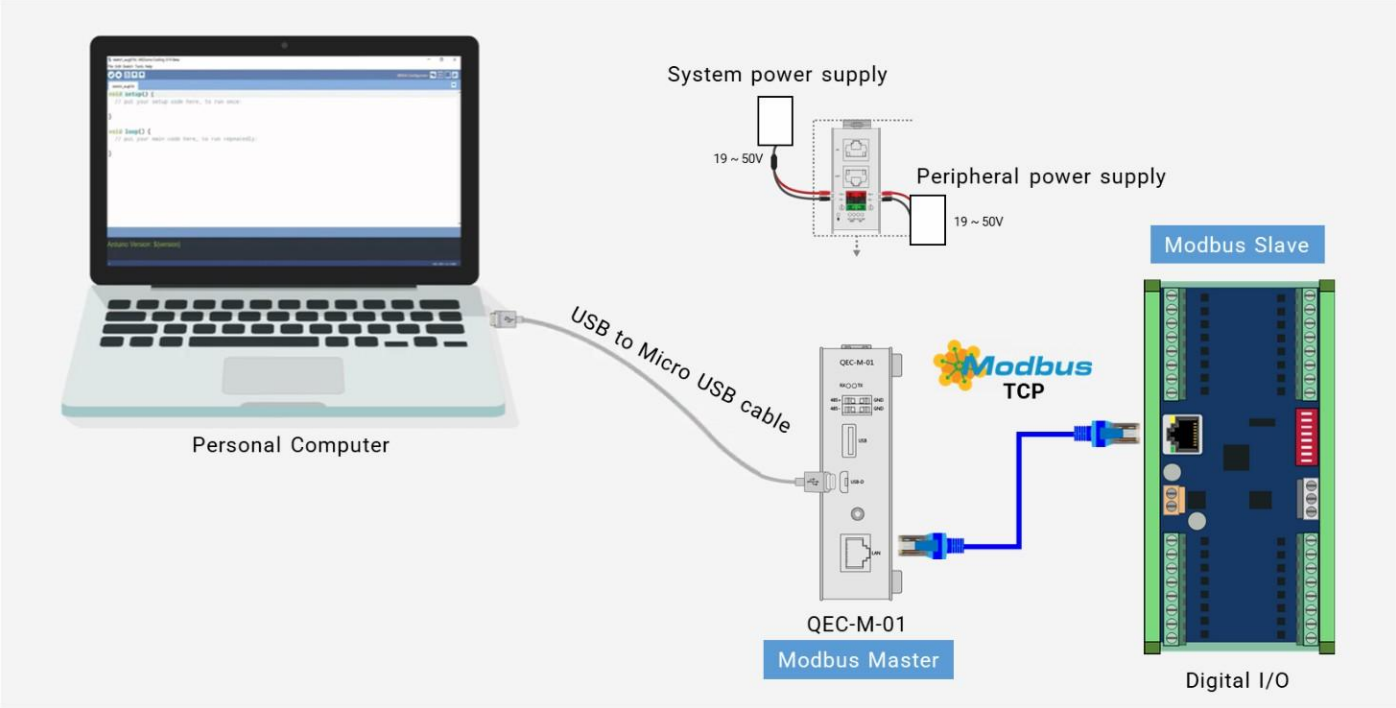
ここでは、以下の機器を使用します。

1. QEC-M-01（EtherCAT M デバイス）
2. Modbus TCP デジタル I/O モジュール（8 入力・8 出力）
3. 24V 電源

1.1 QEC-M-01

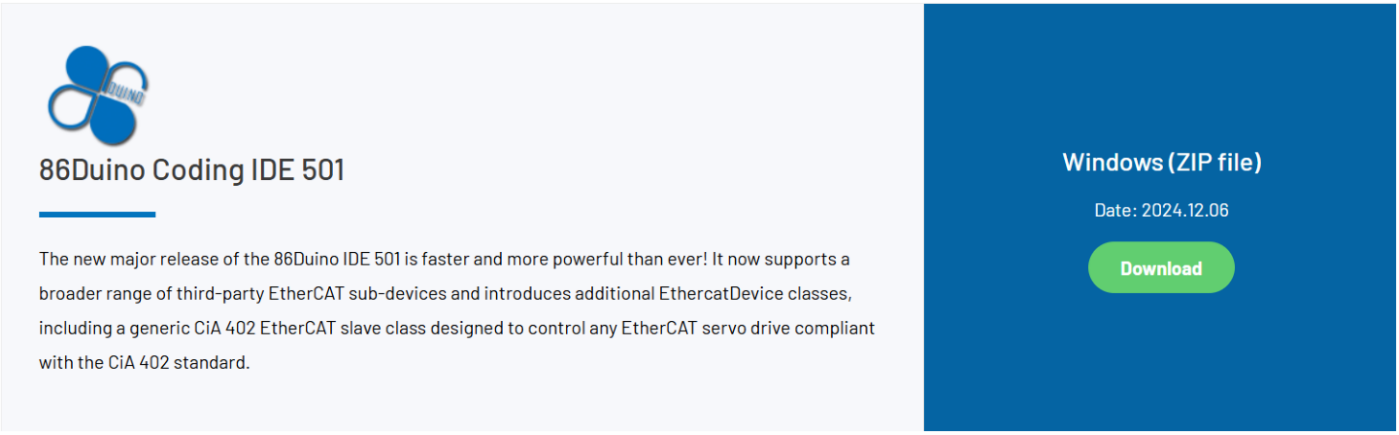
QEC EtherCAT M デバイスです。

1. 電源：
   EU 端子を介して Vs+/Vs- および Vp+/Vp- に 24V 電源を接続します。
2. イーサネット接続：
   前面のイーサネットポートと Modbus I/O モジュールの LAN ポートを RJ45 ケーブルで接続します。



2. ソフトウェア／開発環境

86Duino IDE は https://www.qec.tw/software/ からダウンロードしてください。



ダウンロードした zip ファイルを解凍してください。追加のソフトウェアのインストールは不要で、86duino.exe をダブルクリックするだけで IDE が起動します。



※注意：
Windows に警告が表示された場合は、Details（詳細情報）を一度クリックし、続けて Continue Run（実行）ボタンを一度クリックしてください。

86Duino Coding IDE 501 以降の画面は以下のとおりです。



3. PC との接続と環境設定

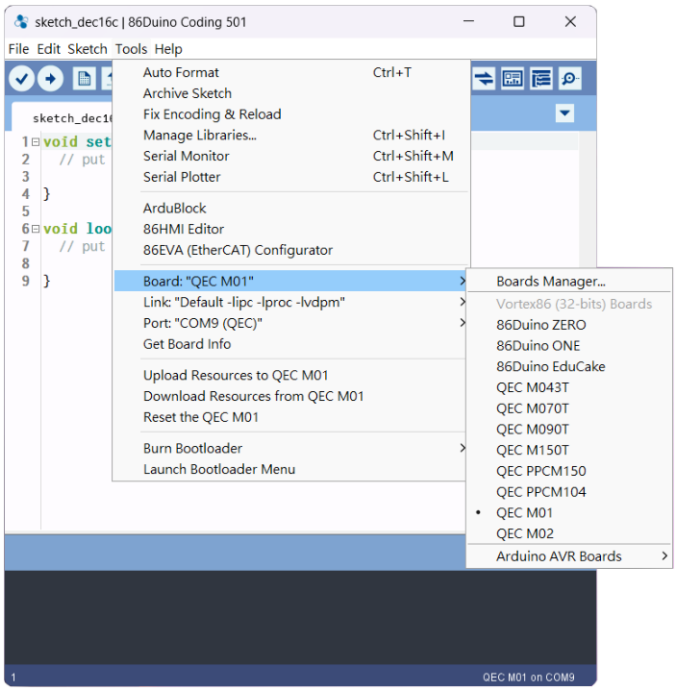
以下の手順で環境を設定します。

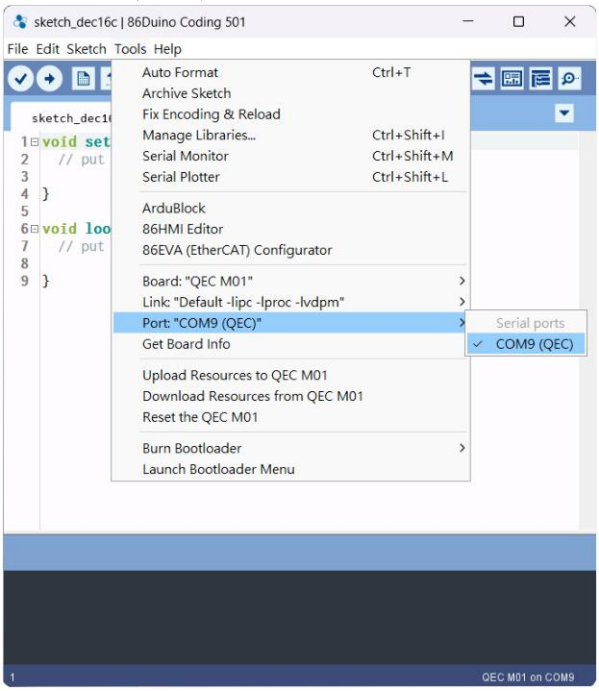
1. Micro USB－USB ケーブルで QEC-M-01 を PC（86Duino IDE インストール済み）に接続します。
2. QEC の電源を入れます。
3. PC で Device Manager（デバイス マネージャー）を開き（Win+X を押して表示されるメニューから選択）、Ports (COM & LPT)（ポート（COM と LPT））を展開します。「Prolific PL2303GC USB Serial COM Port (COMx)」が検出されていることを確認してください。検出されていない場合は、必要なドライバーをインストールしてください。
   （Windows 用 PL2303 ドライバーは こちら からダウンロードできます）





4. 86Duino IDE を開きます。
5. ボードを選択します：IDE のメニューから Tools（ツール）> Board（ボード）> QEC-M-01（または使用する QEC M デバイスの機種）を選択します。
6. ポートを選択します：IDE のメニューから Tools（ツール）> Port（シリアルポート）を選択し、QEC M デバイスが接続されている USB ポート（この例では COM9 (QEC)）を選択します。





4. コードの作成

QEC-M-01 コントローラーは、Modbus TCP プロトコルを介してリモートの DIO（デジタル入出力）モジュールと通信できます。開発には 86Duino IDE と **Ethernet** ライブラリおよび **Modbus** ライブラリを使用します。

Arduino のプログラミングモデルでは、setup() 関数で初期設定を行い、loop() 関数でメインの処理を行います。この例では、1 秒ごとに保持レジスタを読み取り、デジタル出力（コイル）に書き込みます。

ハードウェア構成

- Modbus マスター：QEC-M-01
- Modbus スレーブ：DIO モジュール（IP アドレス 192.168.1.80、Slave ID 28）
- 通信プロトコル：Modbus TCP
- イーサネット MAC/IP：ネットワークアクセス用に QEC-M-01 に割り当て

A. setup 関数

setup() 関数では、Modbus TCP 接続を初期化し、通信の準備を行います。

1. デバッグ用に 115200 baud でシリアル通信を開始します。
2. QEC-M-01 の MAC アドレスとローカル IP アドレスを設定します。
3. スレーブ機器との Modbus TCP 通信を開始します。

B. loop 関数

loop() 関数は 1 秒ごとに以下の処理を実行します。

1. スレーブ機器 28 の保持レジスタ アドレス 4 から 1 ワード（16 ビット）を読み取ります。
2. 読み取った値を使って、アドレス 0 から始まる 8 点のデジタル出力（コイル）を設定します。
3. 読み取ったデータの値を 2 進数でシリアルモニタに出力します。
4. Modbus の読み取りに失敗した場合は、エラーメッセージを出力します。

主な API 関数：

- `readHoldingRegisters(uint8_t slave_id, uint16_t address, uint16_t size, uint16_t *data);`
- `writeMultipleCoils(uint8_t slave_id, uint16_t address, uint16_t size, uint16_t *data);`

サンプルコードは以下のとおりです。

```
#include "Modbus.h"
#include <Ethernet.h>

ModbusMaster bus;

byte mac[] = {0x00, 0x00, 0x00, 0x00, 0x00, 0x00};
IPAddress localIp(192, 168, 1, 101);
IPAddress serverIp(192, 168, 1, 80);

uint16_t data[1];

void setup() {
  Serial.begin(115200);

  Ethernet.begin(mac, localIp);
  bus.begin(MODBUS_TCP, serverIp);
}

void loop()
{
  static uint32_t lastTime = 0;
  if (millis() - lastTime >= 1000) {
    uint8_t result = bus.readHoldingRegisters(28, 4, 1, data);
    bus.writeMultipleCoils(28, 0, 8, &data[0]);
    if (result == 0) {
      Serial.print("Value: "); Serial.println(data[0], BIN);
    } else {
      Serial.println("Error!");
    }
    lastTime = millis();
  }
}
```

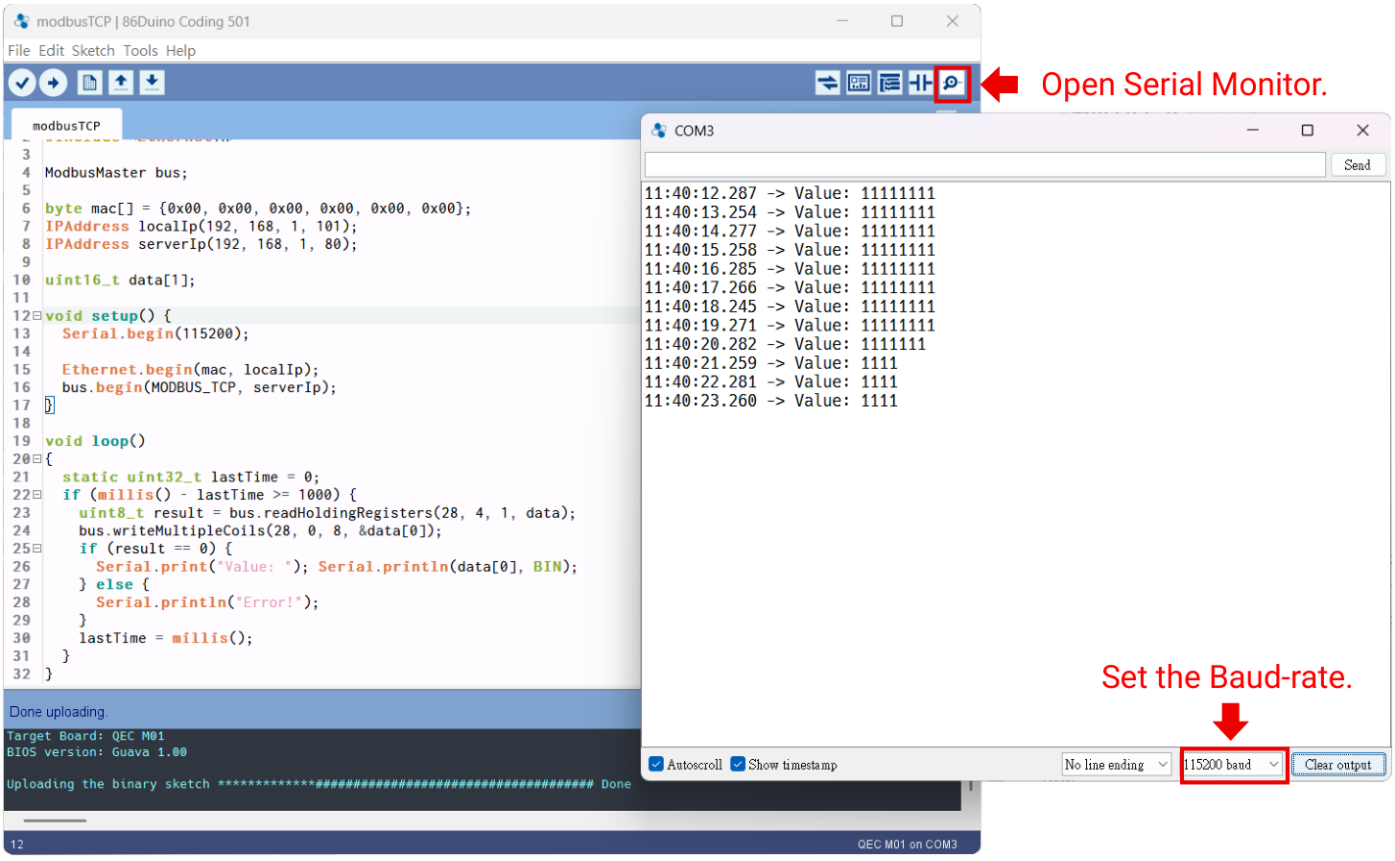
※注意：

コードを書き終えたら、ツールバーの をクリックしてコンパイルします。コンパイルがエラーなく完了したことを確認したら、

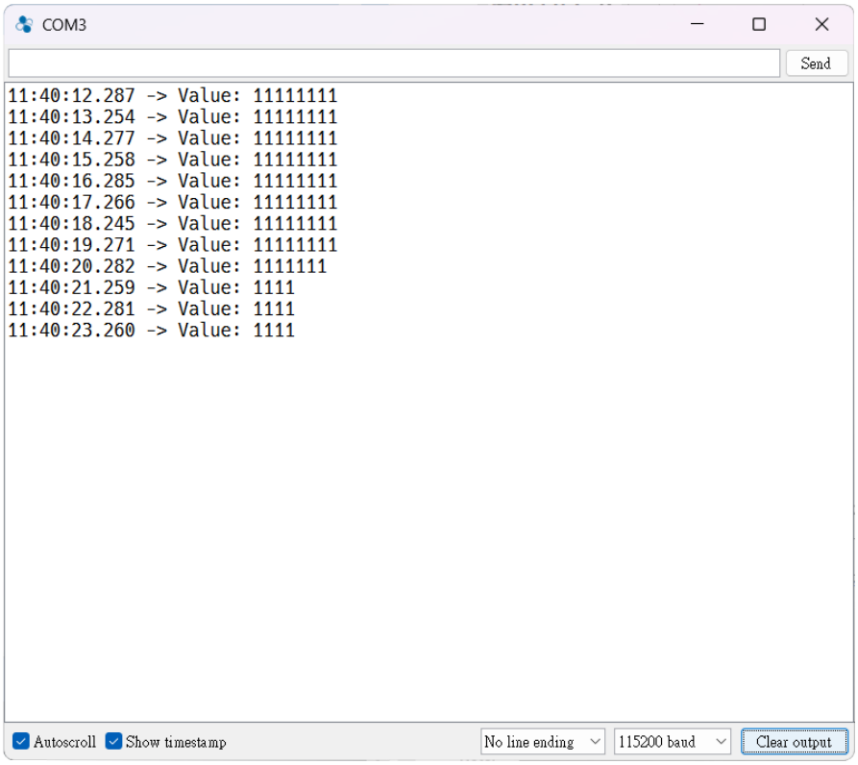
Upload（書き込み）をクリックして書き込みます。



QEC-M-01 へのプログラムの書き込みが完了したら、86Duino IDE の Serial Monitor（シリアルモニタ）を開きます。シリアルモニタのボーレートがプログラムの設定と同じであることを確認してください。



Modbus 通信の設定が正常に完了し、読み書きが始まると、シリアルモニタに「Value: ...」と表示されます。これで、すべての設定が正常に完了したことになります。
シリアルモニタには、Modbus I/O モジュールの現在の入力状態と、制御している出力の状態が表示されます。



保証

本製品は、ご購入日から 1 年間、正常に動作することを保証します。この期間中に本製品が正常に動作しなくなった場合、当社の判断により、以下の条件に定める場合を除き、無償で交換または修理いたします。本保証は、誤用、改造、事故、または災害によって損傷した製品には適用されません。販売元は、本製品の使用、誤用、または使用不能に起因するいかなる損害、逸失利益、逸失した節約、その他の付随的または結果的損害についても責任を負いません。販売元は、その他の関係者による請求についても責任を負いません。返品の際は、事前に販売元から返品承認を得る必要があります。承認は、販売元に電話または FAX で連絡し、返品承認（RMA）番号を申請することで取得できます。返品する製品には、必ず問題の内容を明記した説明書を添付してください。

本書に記載されているすべての商標は、それぞれの所有者の登録商標です。
すべての仕様は予告なく変更されることがあります。
©ICOP Technology Inc. 2025

日本語版資料は、英語版を翻訳したもので、内容に相違が生じる場合には原文を優先します。